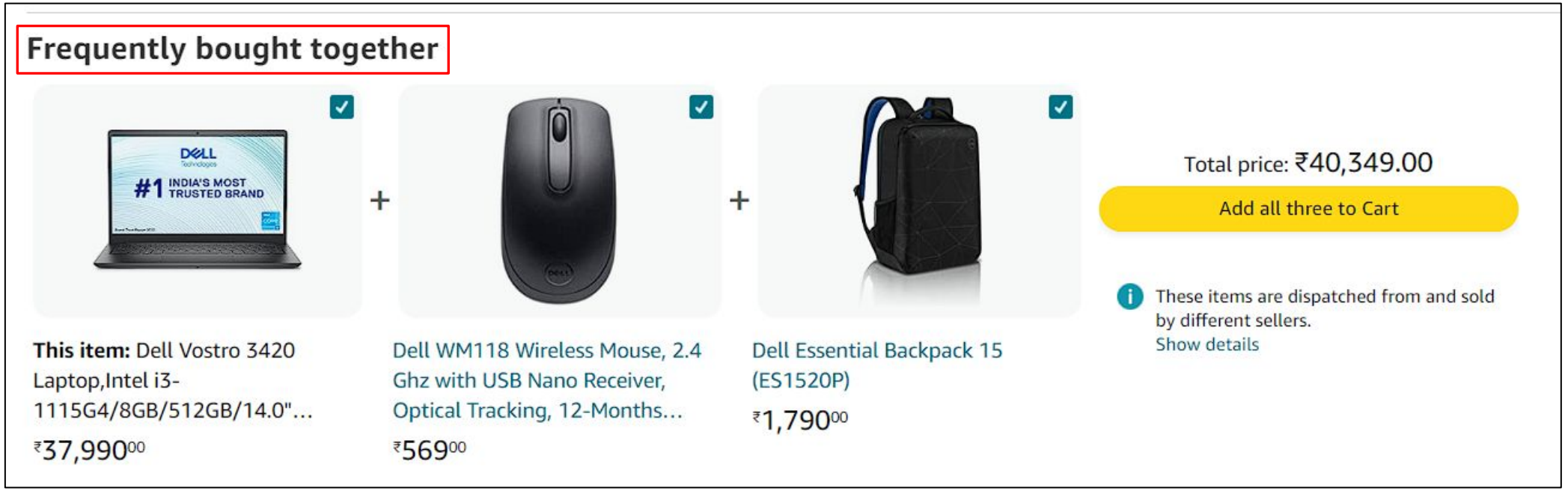
Association Rule Mining

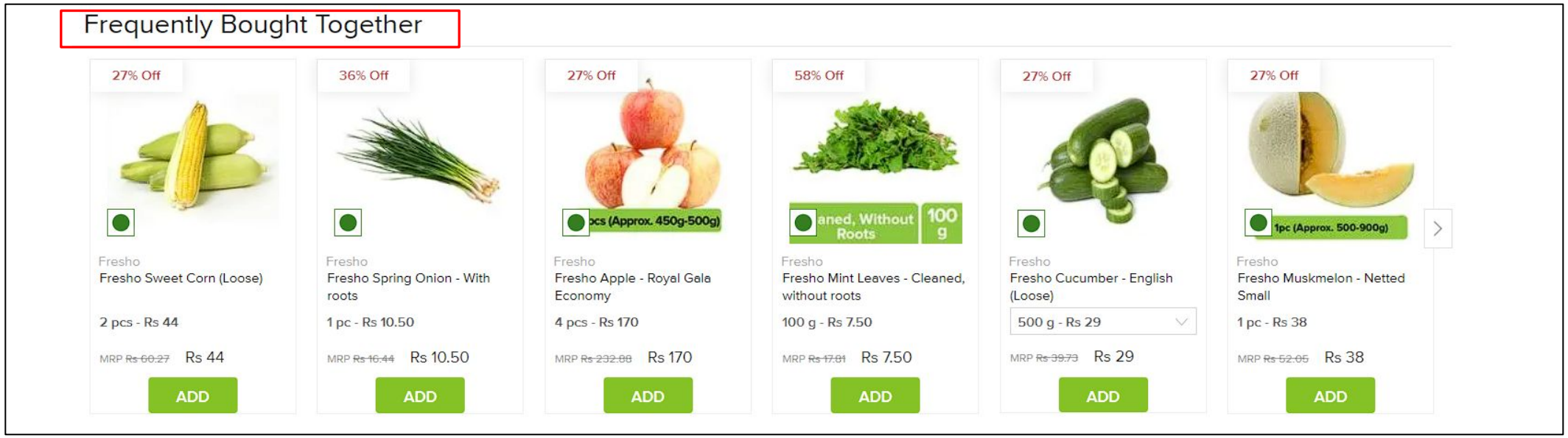
SINDHURA NADENDLA

Association Rule Minining Example

Amazon Items that are frequently bought together



Example of ARM in Big Basket website



Support

Support refers to the default popularity of an item and can be calculated by finding number of transactions containing a particular item divided by total number of transactions. Suppose we want to find support for item B. This can be calculated as:

```
Support(B) = (Transactions containing (B))/(Total Transactions)
```

For instance if out of 1000 transactions, 100 transactions contain Ketchup then the support for item Ketchup can be calculated as:

```
Support(Ketchup) = (Transactions containingKetchup)/(Total Transactions)

Support(Ketchup) = 100/1000
                 = 10%
```

Confidence

Confidence refers to the likelihood that an item B is also bought if item A is bought. It can be calculated by finding the number of transactions where A and B are bought together, divided by total number of transactions where A is bought. Mathematically, it can be represented as:

```
Confidence(A→B) = (Transactions containing both (A and B))/(Transactions containing A)
```

Coming back to our problem, we had 50 transactions where Burger and Ketchup were bought together. While in 150 transactions, burgers are bought. Then we can find likelihood of buying ketchup when a burger is bought can be represented as confidence of Burger -> Ketchup and can be mathematically written as:

```
Confidence(Burger→Ketchup) = (Transactions containing both (Burger and Ketchup))/(Transactions containing A)

Confidence(Burger→Ketchup) = 50/150
                           = 33.3%
```

Lift

`Lift(A -> B)` refers to the increase in the ratio of sale of B when A is sold. Lift(A –> B) can be calculated by dividing `Confidence(A -> B)` divided by `Support(B)`. Mathematically it can be represented as:

```
Lift(A→B) = (Confidence (A→B))/(Support (B))
```

Coming back to our Burger and Ketchup problem, the `Lift(Burger -> Ketchup)` can be calculated as:

```
Lift(Burger→Ketchup) = (Confidence (Burger→Ketchup))/(Support (Ketchup))

Lift(Burger→Ketchup) = 33.3/10
                     = 3.33
```

Lift basically tells us that the likelihood of buying a Burger and Ketchup together is 3.33 times more than the likelihood of just buying the ketchup. A Lift of 1 means there is no association between products A and B. Lift of greater than 1 means products A and B are more likely to be bought together. Finally, Lift of less than 1 refers to the case where two products are unlikely to be bought together.

Steps Involved in Apriori Algorithm

For large sets of data, there can be hundreds of items in hundreds of thousands transactions. The Apriori algorithm tries to extract rules for each possible combination of items. For instance, Lift can be calculated for item 1 and item 2, item 1 and item 3, item 1 and item 4 and then item 2 and item 3, item 2 and item 4 and then combinations of items e.g. item 1, item 2 and item 3; similarly item 1, item2, and item 4, and so on.

As you can see from the above example, this process can be extremely slow due to the number of combinations. To speed up the process, we need to perform the following steps:

1. Set a minimum value for support and confidence. This means that we are only interested in finding rules for the items that have certain default existence (e.g. support) and have a minimum value for co-occurrence with other items (e.g. confidence).
2. Extract all the subsets having higher value of support than minimum threshold.
3. Select all the rules from the subsets with confidence value higher than minimum threshold.
4. Order the rules by descending order of Lift.

Apply Apriori algorithm

The `apriori` class requires some parameter values to work. The first parameter is the list of list that you want to extract rules from. The second parameter is the `min_support` parameter. This parameter is used to select the items with support values greater than the value specified by the parameter. Next, the `min_confidence` parameter filters those rules that have confidence greater than the confidence threshold specified by the parameter. Similarly, the `min_lift` parameter specifies the minimum lift value for the short listed rules. Finally, the `min_length` parameter specifies the minimum number of items that you want in your rules.

Let's suppose that we want rules for only those items that are purchased at least 5 times a day, or 7 x 5 = 35 times in one week, since our dataset is for a one-week time period. The support for those items can be calculated as 35/7500 = 0.0045. The minimum confidence for the rules is 20% or 0.2. Similarly, we specify the value for lift as 3 and finally `min_length` is 2 since we want at least two products in our rules. These values are mostly just arbitrarily chosen, so you can play with these values and see what difference it makes in the rules you get back out.

Thank you